

**REMARKS**

In view of the foregoing amendments and following remarks responsive to the Office Action dated July 10, 2008, Applicants respectfully request favorable reconsideration of this application.

**Matters of Form**

In section 2 of the Office Action, the Office objected to the specification as failing to provide proper antecedent basis for the claimed subject matter. Specifically, the Office asserted that the specification fails to provide antecedent basis for the term "computer readable medium".

In a related rejection, in sections 3 and 4 of the Office Action, the Office rejected claims 16 and 17 under 35 USC 101 asserting that the claimed invention is directed to non-statutory subject matter. Particularly, the Office asserted that claims 16 and 17 recite a "computer readable medium", but the specification fails to provide antecedent basis for this limitation. More particularly, the Office asserted that, without antecedent basis for "computer readable medium" it is unclear if the limitation is intended to be the same as a storage media described as part of the disclosed program product or whether it is intended to be broader than the disclosed storage media. The Office asserted that it believed that it was intended to claim something broader than the disclosed storage media and, particularly, to cover signals, waves and other forms of transmission media that carry instructions.

Applicants have herein amended the preambles of the independent claims 16 and 17 to specify that the computer readable medium is all "computer readable storage medium". This should resolve the issue raised in the rejection under 35 USC 101. Particularly, it seems clear that the Office is looking for language that specifically or cites a storage media, so as to exclude non-statutory computer readable media, such as signal waves and other forms of transmission media that carry instructions. Clearly, the addition of the word "storage" achieves this goal. It is well accepted that the

specification need not provide literal antecedent basis for every word in every claim. Clearly, the storage media that the Office expressly acknowledges is described in the specification is a computer readable storage medium and those of skill in the art know it. Accordingly, there is a proper antecedent basis in the specification for the term "computer readable storage medium".

### **Matters of Prior Art**

The Office also has rejected claims 1-3, 7-9, and 13-17 under 35 USC 102(e) as being anticipated by Schmidt and claims 4-6 and 10-12 under 35 USC 103(a) as being unpatentable over Schmidt in view of Factor. These are the same rejections as asserted in the previous Office Action.

In response, Applicants argued that Schmidt fails to disclose removing from a JAR file any information unnecessary for executing a particular application. The Office responded negatively to this argument asserting that the claims recite processing a Java archive file to provide an interpretable application file and that Schmidt discloses processing a Java archive file (column 9, line 65-column 10, line 22) to create an interpretable application file (column 10, lines 46-60). The Office argues that Schmidt teaches creating a synthesized archive file by executing a sequence of copy and/or delete operations on the entries in either the original archives, on the entries in the difference archive file, and/or on the entries in the synthesized archive file in progress (column 12, line 60-column 13, line 16).

The present invention pertains to a process for producing compressed Java jar files including byte code which remains directly interpretable by a Java virtual machine process comprising mapping at least one of application defined class, field, and method names to shorter names, and mapping at least one of target environment defined class, field, and method names to corresponding target device names.

The Office's primary reference, Schmidt, has nothing to do with this. Rather, Schmidt pertains to a technique by which servers can update Java applets at clients by transmitting a "difference" file from a server to a client, rather than sending an entire

new Java jar file. In accordance with the process disclosed in Schmidt, the server has an "original" file that is the file it originally sent to the client. When that original Java jar file is modified for whatever reason, the server creates a new file, which Schmidt terms a "target" file. The server compares the original file to the target file and generates a difference file. The difference file contains (1) any new files that were neither in the original file nor could be duplicated from another file in the original file and (2) a series of copy and/or delete instructions. The server sends the difference file to the client machine, rather than an entire new Java jar. When the client receives a difference file, it copies any new files contained in the difference file and then performs the various copy and/or delete instructions contained in the difference file in order to convert its original copy of the Java jar file into a new file, called a "synthesized" file, that is identical for substantive purposes to the aforementioned target file (although possibly not perfectly identical in form).

The Office has focused on the delete commands that might be contained in a difference file and asserted that the language in the claims of "removing from said JAR file at least a portion of information not necessary for executing said application" reads on the simple act in Schmidt of deleting a file within a Java jar file.

Without regard to the actual language of the claims, it should be clear that Schmidt discloses something radically different from Applicants' invention. In particular, in Schmidt, the files that are deleted are deleted because they no longer form part of the application because the application has been changed. In other words, the removal of the files changes the functionality of the application in Schmidt. In the present invention, on the other hand, the removed information is removed to compress the file without changing the functionality of the application.

Applicants have herein amended the language of independent claims 1, 14, 15, 16, and 17 in order to more clearly distinguish over Schmidt. Particularly, Applicants have added at the end of each relevant clause of claims 1, 14, 16, and 17 the phrase "to compress said JAR file". The amendment to claim 15 is different due to the different

form of that claim. The amended claim language is supported in the specification on at least, page 3, lines 24-28.

This language should sufficiently distinguish over the very different Schmidt reference insofar as Schmidt does not produce a compressed Java jar file. Rather, Schmidt just produces a different Java jar file with different functionality.

Furthermore, Applicants traverse the Office's assertion that Schmidt discloses the second step of independent claims 1, 14, 16, and 17 of "mapping at least one of application defined interface, class, field and method names to shorter names". Particularly, the Office asserts that this is found in Schmidt at column 12, lines 1-17.

However, this is not true. This portion of the Schmidt instead discusses the differences between the original file and the target file at the server. It does not relate to the synthesized file created at the client. Furthermore, it is merely an explanation of the differences between the original file and the target file. It serves as a preface to the discussion of how the difference file will represent what those differences are so that the client can modify its Java jar file accordingly to create the synthesized file that is a reproduction of the target file. It simply does not have anything to do with mapping names to shorter names.

Even further, Applicants traverse the Office's assertion that Schmidt discloses the third step of independent claims 1, 14, 16, and 17 of "mapping at least one of target environment defined interface, class, field and method names to corresponding target device names". Particularly, the Office asserts that this is found in Schmidt at column 12, lines 18-35.

However, this also is not true. This portion of Schmidt also discusses the differences between the original file and the target file at the server. It does not relate to the synthesized file at the client. Furthermore, it is merely an explanation of the fact that the target file may contain entries that are new as compared to the original file and that it is possible that some of the new files may actually have identical content to some of the old files. It serves as a preface to the discussion of how the difference file will represent what those differences are so that the client can modify its Java jar file

accordingly to reproduce the target file. It simply does not have anything to do with mapping names to shorter names. Particularly, in a later discussion in Schmidt of the difference file, Schmidt points out that, when a new file is identical to an old file with a different name, Schmidt avoids the need to transmit the new file by instead including an instruction in the difference file to copy the identical old file into a new file with the new name. This does not have anything to do with step 3 of claim 1 (or any of the other similar steps recited in independent claims 14, 16, and 17).

Accordingly, claims 1, 14, 15, 16, and 17 clearly patentably distinguish over Schmidt.

All of the dependent claims 2-13 patentably distinguish over Schmidt for at least all the same reasons. With respect to dependent claims 4-6 and 10-12, which depend from claim 1 and were rejected as obvious over Schmidt in view of Factor, Factor does not provide the above-noted teachings lacking from Schmidt.

Furthermore, the dependent claims recite even further distinguishing features over the prior art of record. For instance, claim 7 depends from claim 1 and adds "preferentially remapping application references to at least one of target environment defined interface, class, field and method names". The Office asserted that this is found at column 12, lines 18-35 of Schmidt. However, as just explained in connection with step 3 of claim 1, this simply is not true.

Furthermore, claim 8 depends from claim 1 and adds that "target environment obfuscation is provided in which symbols used in the target environment are replaced with shorter names". The Office asserted that this is found in column 12, lines 3-11 of Schmidt. However, Applicants just explained above in connection with the step 2 of claim 1 that this portion of Schmidt relates to a very different topic, namely, it is merely an explanation of the differences between the original file and the target file. It serves as a preface to the discussion of how the difference file will represent what those differences are so that the client can modify its Java jar file accordingly to reproduce the target file. It simply does not have anything to do with mapping names to shorter names.

Claim 9 depends from claim 1 and adds that "an application obfuscation is provided in which symbols used in said application or replaced with shorter names that do not overlap the target device names used for target environment obfuscation". Again, the Office relies on column 12, lines 11-17 in Schmidt as teaching this. However, as noted above, this portion of Schmidt does not have anything to do with using shorter names.

Claim 13 depends from claim 1 and adds that "the mapping steps are only used for mapping private symbols". The Office asserts that this is found in column 12, lines 18-35. However, Applicants already explained what that portion of Schmidt pertains to and how it does not disclose the claimed mapping steps. Accordingly, it obviously cannot disclose the qualification of the mapping step recited in claim 13 either.

Turning to independent claim 15, in addition to distinguishing over Schmidt because it now recites that the process compresses the file as noted above, it even further distinguishes over the prior art of record for its own unique reasons. Specifically, claim 15 recites "iteratively solving application defined target environment defined class field and method names to interpret application byte codes presented within a round Java archive (JR) file to produce a minimized jar file". The Office asserted that this is found in Schmidt at column 12, line 60-column 13, line 16 and column 15, line 11 to column 16, line 29.

Applicant respectfully traverses. To start with, there are no ground Java archive files in Schmidt. Secondly, the cited portions of Schmidt disclose nothing like what is claimed in claim 15. Particularly, column 15, line 11 through column 16, line 29 describes the flowchart of Figure 11, which pertains strictly to how to generate the difference file. Particularly, it describes a process of starting with the original and target files (steps 732 and 734), determining the differences between those two files (steps 736 and 738), adding any new files and their contents to the difference file (step 740), adding appropriate copy commands to the difference file with respect to files that have the same contents as old files but different names (steps 736, 738, and 742), and deleting whatever is left over at the end (steps 748). (The flowchart also includes two

other steps, steps 744 and 746, but they merely make the process iterative for each entry in the target file).

This process is essentially irrelevant to the subject matter of claim 15. It certainly does not disclose "solving application defined and target environment defined class, field and method names" and has nothing to do with interpreting application byte codes".

### **New Claims**

Applicant has added new claims 18-20 to claim various specific techniques for removing information from a JAR file discussed in the specification, namely, archive tersing (page 8, line 16-page 9, line 28 of the specification), class tersing (page 9, line 30-page 11, line 2 of the specification), and opcode replacement (page 11, line 4-page 12, line2).

### **Conclusion**

The present invention is not taught or suggested by the prior art. Accordingly, the Examiner is respectfully requested to reconsider and withdraw the rejection of the claims. An early Notice of Allowance is earnestly solicited.

The Commissioner is hereby authorized to charge any additional fees or credit any overpayment associated with this communication to Deposit Account No. 50-4364.

In view of the foregoing remarks, this application is now in condition for allowance. Applicants respectfully request the Office to issue a Notice of Allowance at the earliest possible date. The Examiner is invited to contact Applicants' undersigned counsel by telephone call in order to further the prosecution of this case in any way.

Respectfully submitted,

Dated: September 15, 2008

/Theodore Naccarella/  
Theodore Naccarella, Reg. No. 33,023  
SAUL EWING LLP  
Centre Square West  
1500 Market Street, 38<sup>th</sup> Floor  
Philadelphia, PA 19102-2189  
Telephone: 215 972 7877  
Email: TNaccarella@saul.com

TXN:rb